Outline

1. Introduction
2. Background on key concepts
3. The SWHID identifier
4. Turning SWHID into a publicly available specification
5. The road ahead
<table>
<thead>
<tr>
<th>Objective</th>
<th>bring you all up to speed on the SWHID specification effort, and kickstart the work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>- duration: 1 hour (45m plus 15m Q&amp;A); write questions in the chat</td>
</tr>
<tr>
<td></td>
<td>- session will be recorded</td>
</tr>
<tr>
<td></td>
<td>- set your Zoom id to your full name</td>
</tr>
<tr>
<td>Agenda</td>
<td>- recall basic notions of identifiers, then focus on the SWHID</td>
</tr>
<tr>
<td></td>
<td>- review of the way SWHIDs are composed and computed</td>
</tr>
<tr>
<td></td>
<td>- focus on the key parts of the specification that need work</td>
</tr>
<tr>
<td></td>
<td>- guided tour of the contribution and editorial process</td>
</tr>
<tr>
<td></td>
<td>- governance and licensing</td>
</tr>
</tbody>
</table>
1. Introduction
2. Background on key concepts
3. The SWHID identifier
4. Turning SWHID into a publicly available specification
5. The road ahead
Identification vs Location

Identification of a book

Goal: identify a book
- one ISBN number per published book
- ISO 2108 standard specification

Location of (a copy of) a book

Goal: find (a copy of) a book
- many locations (locations can change!)
- many approaches for call numbers

we are interested in **identification**, not in location
Extrinsic vs Intrinsic identifiers

In a nutshell (for more info see this dedicated blog post)

Main difference: how the relation between identifier and designated object is created and maintained. Persistence is a key desired property.

<table>
<thead>
<tr>
<th>Extrinsic</th>
<th>Intrinsic</th>
</tr>
</thead>
<tbody>
<tr>
<td>relation</td>
<td>register</td>
</tr>
<tr>
<td>persistence</td>
<td>external $^a$</td>
</tr>
<tr>
<td></td>
<td>convention</td>
</tr>
<tr>
<td></td>
<td>internal</td>
</tr>
<tr>
<td>pre-internet</td>
<td>passport number, ISBN, SSN, etc.</td>
</tr>
<tr>
<td></td>
<td>Music/Chemistry notations</td>
</tr>
<tr>
<td></td>
<td>e.g. NaCl is table salt</td>
</tr>
<tr>
<td>internet era</td>
<td>DOI, Handle, Ark, etc.</td>
</tr>
<tr>
<td></td>
<td>cryptographic hashes</td>
</tr>
<tr>
<td></td>
<td>e.g.: git, bitcoin, SWHID</td>
</tr>
</tbody>
</table>

$^a$"persistence... is a function of administrative care" RFC 3650 (Handle System Overview, 2003)

Here we are interested in normalising the SWHID intrinsic identifier
Outline

1. Introduction
2. Background on key concepts
3. The SWHID identifier
4. Turning SWHID into a publicly available specification
5. The road ahead
Bird’s eye view of the SWHID *Intrinsic* Identifier

Structure of a SWHID identifier

```
swrl:cnt:41ddb23118f92d7218099a5e7a990cf58f1d07fa
```

Current status

30+B SWHIDs in the Software Heritage archive

Mention in Linux Foundation’s SPDX 2.2; IANA registered; WikiData P6138
SWHID computation: a worked example

Contents

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that prevent others from doing what will break these conditions. This license is designed to preserve your权利, not to deny them.

sha1: 8624bcdae55baeef...
sha256: 8ceb4b9ee5aded...
sha1_git: 94a9ed024d385...
length: 35147
SWHID computation: a worked example
Directories

id: 515f00d44e92c65322aaa9bf3fa097c00ddb9c7d

?
SWHID computation: a worked example

Directories

100644 blob c5baade4c44766042186ef858c0fd63d587ebf09 .gitignore
100644 blob 2d0a34af6f52cf3cf6b8c2f7bd6048fbd255e77f AUTHORS
100644 blob 94a9ed824d3859793618152e559a168bbcb5b5e LICENSE
100644 blob d9b265a435a43f8a79a84e0867751dfb095c7bb MANIFEST.in
100644 blob 524175c2bad0b35b975f79284c2f5a6d5eaf2eb4 Makefile
100644 blob 5c7e3a5bbdbd938682ba7793f440492ed9678bb3 Makefile.local
100644 blob 8617980629cd24a6880484f99a749b085b3e07b README.db.testing
100644 blob 76b29f94cf8150e069c414d38d78d7ce08ec514e README.dev
040000 tree e1e1e0cecf948af0b93adb0372afa89f12e92618a bin
040000 tree 83e56d0beaf7793c77a45a345c80fcb8af503013 debian
040000 tree a34c9c4ba213f9eced67f9816348b2795557af5 docs
100644 blob f2a6d32c6135aa7287bbd76167b01df2ae4f1539 requirements.txt
100755 blob eee147c36caf1bcbc2d820da8d0c926cb5b681800c setup.py
040000 tree 224bb4c1f4c67fca1d160bfdf2d06094e7e1abf3 sql
040000 tree 8631c9cd77bee993168187ab5ba951f40c63008e swh
040000 tree 8fb905b56ba8ed692f1209b2773b474c6c1d66c1 utils

id: 515f00d44e92c65322aa9bf3fa097c00ddb9c7d
SWHID computation: a worked example
Revisions

<table>
<thead>
<tr>
<th>Details</th>
<th>Changes</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHA: 953634dca6ba5e37e3ee426ba091092c267f9f6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Author: Nicolas Dandrimont <a href="mailto:nicolas@dandrimont.eu">nicolas@dandrimont.eu</a> (Thu Sep 1 14:26:13 2018)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Committer: Nicolas Dandrimont <a href="mailto:nicolas@dandrimont.eu">nicolas@dandrimont.eu</a> (Thu Sep 1 14:26:13 2018)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subject: provenance.tasks: add the revision -&gt; origin cache task</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parent: 515f00d44e92c65322aaa9bf3fa097c00d9c7d</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Author: Nicolas Dandrimont <a href="mailto:nicolas@dandrimont.eu">nicolas@dandrimont.eu</a> 1472732773 +0200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Committer: Nicolas Dandrimont <a href="mailto:nicolas@dandrimont.eu">nicolas@dandrimont.eu</a> 1472732773 +0200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>id: 963634dca6ba5dc37e3ee426ba091092c267f9f6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tree** 515f00d44e92c65322aaa9bf3fa097c00d9c7d
**parent** fc3a8b59ca1df424d860f2c29ab07fee4dc35d10
author Nicolas Dandrimont <nicolas@dandrimont.eu> 1472732773 +0200
committer Nicolas Dandrimont <nicolas@dandrimont.eu> 1472732773 +0200

provenance.tasks: add the revision -> origin cache task
SWHID computation: a worked example
SWHID computation: a worked example

Release

object c0c9f16b1e134f593e7567570a1761b156e6eb1d

type commit
tag v0.0.51
tagger Nicolas Dandrimont <nicolas@dandrimont.eu> <1472042163 +0200>

Release swh.storage v0.0.51

- Add new metadata column to origin_visit
- Update swh-add-directory script for updated API

...END PGP SIGNATURE.....

id: 85083a5cc14a441c89dea73f5bf67c3f9c6afdb

---BEGIN PGP SIGNATURE-----

c0c9f16b1e134f593e7567570a1761b156e6eb1d

---END PGP SIGNATURE-----
WHID computation: a worked example
SWHID computation: a worked example

Snapshots

commit 08ffe2b5770109525eb3ce21691466c53a1d3158 refs/heads/alive
commit ba4a3a24e3f9e223e3e292b0e2c6e4f6ce61c679b refs/heads/directory-listing-arrays
commit d6e9e0d8f928238f6f589b27fe1c85d2723289d6c5 refs/heads/too
commit c77ff9eaa8e6b227b04690f5f508019f67e45be88 refs/heads/master
commit 7ec3a197cf6c6d282404e3a5b1ed9e8a4b43616a0f32 refs/heads/tmp-directory-add
commit 642a28b5f37de850b0a5d42b530ee4fba2252e82e refs/heads/tmp/generic-releases
tag 21f043b137fc7f68a9665779f7f4690c757f755 refs/tags/v0.1.0
[tag 72a2199a3a80e43999d080db867fbf8bbee72aee2cda]
tag 3090e43a8e8b8b78e8b53767f823b8fbf4a1f3c refs/tags/v0.1.3
[tag 333e742b74a93b5a56a67777b8d58f6674f6bc6555 refs/tags/v0.1.12]
tag 067f465275b327cf5903112c2fa936c3f3b43035d refs/tags/v0.1.13
[tag 5a635d2f8e8d28f9a5b2d87442667921a1e32f3bd refs/tags/v0.1.14]
tag 586f4be45bb4b5f0ba085f9367643c3bcba1ac77f refs/tags/v0.1.15
[tag 8c06b8854f99bf3f331777428f26f69d6e5b51c refs/tags/v0.1.16]
tag a042a4de3c39fbec035ff28f4eb33c899abc2d6 refs/tags/v0.1.17
[tag 228a2f16550d12222e5655945d2e1e06fc49939d refs/tags/v0.1.18]
tag 609679a4ec5d4977c0d2aadd09b0e62535e1f7c refs/tags/v0.1.19
[tag 32b75a59f42a328baeda651a10a03d3822c27a2e7 refs/tags/v0.1.20]
tag 3147c3d31c4c6d6f492f881e900b1237ebd0ff2c7 refs/tags/v0.1.20
[tag 215e58daba1b11e86be6b7e2e64b6673a87908 refs/tags/v0.1.21]
tag 3b1e9c27a28d6722120a1aebd4cf85f0e8e refs/tags/v0.1.22
[tag 8c06b8854f99bf3f331777428f26f69d6e5b51c refs/tags/v0.1.23]

id: b464cad1b66fff266a37b46ea6e7a04b545e904b
Outline

1. Introduction
2. Background on key concepts
3. The SWHID identifier
4. Turning SWHID into a publicly available specification
5. The road ahead
First step: the name of the game

Origin of the SWHID acronym

SWHIDs were born as
- intrinsic identifiers
- designed for
  - software source code
  - archived in Software Heritage

Hence the acronym:

\[
\begin{align*}
\text{SW} & \quad \text{Software} \\
\text{H} & \quad \text{Heritage} \\
\text{ID} & \quad \text{IDentifier}
\end{align*}
\]

Proposed reformulation

SWHIDs are
- based on a cryptographic hash
- can be used independently of the Software Heritage Archive,
- not restricted to source code

Hence the proposal:

\[
\begin{align*}
\text{SW} & \quad \text{Software} \\
\text{H} & \quad \text{Heritage Hash} \\
\text{ID} & \quad \text{IDentifier}
\end{align*}
\]
Setting the stage

Our goal

Create a specification that
- is complete, precise and non ambiguous (pictures are simplified representations!)
- allows any "person skilled in the art" to implement the same calculation algorithm

To this end we need to get right:
- five key parts in the core: cnt, dir, rel, rev, snp
- qualifiers: easier, but important too
- reference to external standards used (e.g. SHA algorithm)

What we have

- high level documentation from Software Heritage
- two complete implementations (one in Python for SWH, one in OCaml for Opam)
- a draft specification that needs to be completed
Governance and License

Bylaws
Detailed in the Governance Document
- consensus based decision making
- open process to produce an Approved Specification

License
specification Community Specification License
contributions specific CLA accepted by sign-off on GitHub

Coordination by the Core Team
Alexios Zavras, Jean-François Abramatic, Morane Gruenpeter,
Roberto Di Cosmo, Stefano Zacchioli
Roles: maintainers, editors
Documents, collaboration tools and workflow

**Specification document**

- **sources** GitHub repository https://github.com/swhid/specification
- **rendered** swhid.org website

**Working on the specification**

- **contribution workflow** based on **issues** and **pull requests** on GitHub
  - N.B: pull requests **must be signed-off**
- editors **merge** pull requests **on consensus**
- merges **rebuild** the **rendered version** on swhid.org

**Mailing list: swhid-discuss**

used mostly for coordination by the editors
(e.g. scheduling meeting, votes, version freeze, etc.)
Working with issues and pull requests

- the directory SWHID
  - issue
  - pull request

Taking a vote

- the name of the game poll is open at https://bit.ly/swhid-name-poll
1. Introduction
2. Background on key concepts
3. The SWHID identifier
4. Turning SWHID into a publicly available specification
5. The road ahead
Ready, set, go!

Timeline

- **phase 1** complete and accurate v1.0 as soon as possible (end of April desirable)
  - help establish the whole process
  - cover *only* what is already known and being used
  - focus on items labeled **blocker**, make them *complete*, *precise* and *non ambiguous*

- **phase 2** work towards v1.1
  - handle other feedback / input / requests
  - get the version candidate to become an ISO standard

Questions?

Links

- main entry point: [https://swhid.org](https://swhid.org)
- specification sources: [https://github.com/swhid/specification](https://github.com/swhid/specification)